Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2003-23

A GUIDE TO TABU SEARCH

by

Michel Gendreau
Jean-Yves Potvin

September 2003

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone: (514) 343-7575 / Télécopieur: (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

104-00252/

PUBLICATION
CRT-2003-23

A GUIDE TO TABU SEARCH

by

Michel Gendreau[1]
Jean-Yves Potvin[1]

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR).

1. Département d'informatique et de recherche opérationnelle et Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7

Département d'informatique et de recherche opérationnelle - Publication #1242

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2003

Abstract

This paper presents the fundamental concepts of tabu search (TS) in a tutorial fashion. Special emphasis is put on showing the relationships with classical Local Search methods and on the basic elements of any TS heuristic, namely, the definition of the search space, the neighborhood structure, and the search memory. Other sections cover other important concepts such as search intensification and diversification and provide references to significant work on TS. Recent advances in TS are also briefly discussed.

Keywords: tabu search; introduction; tutorial.

Résumé

Cet article présente les concepts fondamentaux de la méthode de recherche avec tabous (RT) sous la forme d'un exposé didactique. Une attention toute particulière est accordée aux relations existant entre cette méthode et les approches classiques de recherche locale, ainsi qu'aux éléments de base de n'importe quelle heuristique RT, à savoir, la définition de l'espace de recherche, la structure de voisinage et la mémoire de la recherche. D'autres sections traitent d'autres concepts importants comme l'intensification et la diversification de la recherche et fournissent des références à des ouvrages-clés en RT. Les travaux récents en RT font aussi l'objet d'une courte section.

Mots-clés : recherche avec tabous; introduction; exposé didactique.

1. INTRODUCTION

Over the last fifteen years, hundreds of papers presenting applications of tabu search, a heuristic method originally proposed by Glover (1986), to various combinatorial problems have appeared in the operations research literature (see, for example, Glover and Laguna, 1997, Glover et al., 1993, Osman and Kelly, 1996, Pardalos and Resende, 2002, Ribeiro and Hansen, 2002, Voss et al., 1999). In several cases, the methods described provide solutions very close to optimality and are among the most effective, if not the best, to tackle the difficult problems at hand. These successes have made tabu search extremely popular among those interested in finding good solutions to the large combinatorial problems encountered in many practical settings. Several papers, book chapters, special issues and books have surveyed the rich tabu search literature (a list of some of the most important references is provided at the end). In spite of this abundant literature, there still seem to be many researchers who, while they are eager to apply tabu search to new problem settings, find it difficult to properly grasp the fundamental concepts of the method, its strengths and its limitations, and to come up with effective implementations. The purpose of this paper is thus to focus on the fundamental concepts of tabu search. Throughout the paper, two relatively straightforward, yet challenging and relevant, problems will be used to illustrate these concepts: the job shop scheduling problem and the capacitated plant location problem. These will be introduced in the following section.

2. ILLUSTRATIVE PROBLEMS

2.1 The job-shop scheduling problem

The job shop scheduling problem is one of the most studied problems in combinatorial optimization and a large number of papers and books deal with the numerous procedures that have been proposed to solve it, including several tabu search implementations. Although a large number of variants are found in the literature (and even more in the real world), the "classical" problem can be stated as follows. We first assume that n jobs must be scheduled on m machines. Each job corresponds to a fixed sequence of m operations, one per machine, where each operation must be processed on a specific machine for a specified duration. Note that the processing order on the machines does not need to be the same from one job to another. Each machine can process at most one operation at a time and, once started, an operation must be completed without interruption. The goal is to assign operations to time slots on the machines in order to minimize the maximum completion time of the jobs, which is also known as the makespan. A solution to this problem can be seen as a set of m permutations of the n jobs, one for each machine, with the associated machine schedules (Anderson, Glass and Potts, 1997).

2.2 The capacitated plant location problem

The capacitated plant location problem is one of the basic problems in location theory. It is encountered in many application settings that involve locating facilities with limited capacity to provide services. The problem can be formally described as follows. A set of customers I with demands d_i, $i \in I$, for some product are to be served from plants located in a subset of sites from a

given set J of "potential sites". For each site $j \in J$, the fixed cost of "opening" the plant at j is f_j and its capacity is K_j. The cost of transporting one unit of the product from site j to customer i is c_{ij}. The objective is to minimize the total cost, i.e., the sum of the fixed costs for open plants and the transportation costs.

Letting x_{ij} ($i \in I$, $j \in J$) denote the quantity shipped from site j to customer i (the x_{ij}'s are the so-called *flow variables*) and y_j ($j \in J$) be a 0-1 variable indicating whether or not the plant at site j is open (the y_j's are the location variables), the problem can be formulated as the following mathematical program:

$$\text{Minimize } z = \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\text{subject to } \sum_{j \in J} x_{ij} = d_i, i \in I$$

$$\sum_{i \in I} x_{ij} \le K_j y_j, j \in J$$

$$x_{ij} \ge 0, i \in I, j \in J$$

$$y_j \in \{0,1\}, j \in J$$

Remark 1. For any vector $\tilde{y}$ of location variables, optimal (w.r.t. to this plant configuration) values for the flow variables $x(\tilde{y})$ can be retrieved by solving the associated transportation problem:

$$\text{Minimize } z(\tilde{y}) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\text{subject to } \sum_{j \in J} x_{ij} = d_i, i \in I$$

$$\sum_{i \in I} x_{ij} \le K_j \tilde{y}_j, j \in J$$

$$x_{ij} \ge 0, i \in I, j \in J$$

If $\tilde{y} = y^*$, the optimal location variable vector, the optimal solution to the original problem is simply given by $(y^*, x(y^*))$.

Remark 2. An optimal solution of the original problem can always be found at an extreme point of the polyhedron of feasible flow vectors defined by the constraints:

$$\sum_{j \in J} x_{ij} = d_i, i \in I$$

$$\sum_{i \in I} x_{ij} \leq K_j, j \in J$$

$$x_{ij} \geq 0, i \in I, j \in J .$$

This property follows from the fact that the capacitated plant location problem can be interpreted as a fixed-charge problem defined in the space of the flow variables. This fixed-charge problem has a concave objective function that always admits an extreme point minimum. The optimal values for the location variables can easily be obtained from the optimal flow vector by setting y_j equal to 1 when $\sum_{i \in I} x_{ij} > 0$, and to 0 otherwise.

3. BASIC CONCEPTS

3.1 Historical background

Before introducing the basic concepts of tabu search, we believe it is useful to go back in time to try to better understand the genesis of the method and how it relates to previous work.

Heuristics, i.e., approximate solution techniques, have been used since the beginnings of operations research to tackle difficult combinatorial problems. With the development of complexity theory in the early 70's, it became clear that, since most of these problems were indeed NP-hard, there was little hope of ever finding efficient exact solution procedures for them. This realization emphasized the role of heuristics for solving the combinatorial problems that were encountered in real-life applications and that needed to be tackled, whether or not they were NP-hard. While many different approaches were proposed and experimented with, the most popular ones were based on hill climbing. The latter can roughly be summarized as an iterative search procedure that, starting from an initial feasible solution, progressively improves it by applying a series of local modifications or moves (for this reason, hill climbing is in the family of local search methods). At each iteration, the search moves to an improving feasible solution that differs only slightly from the current one. In fact, the difference between the previous and the new solution amounts to one of the local modifications mentioned above. The search terminates when no more improvement is possible. At this point, we have a local optimum with regard to the local modifications considered by the hill climbing method. Clearly, this is an important limitation of the method: unless one is extremely lucky, this local optimum will often be a fairly mediocre solution. The quality of the solution obtained and computing times are usually highly dependent upon the "richness" of the set of transformations (moves) considered at each iteration.

In 1983, a new heuristic approach called simulated annealing (Kirkpatrick, Gelatt and Vecchi, 1983) was shown to converge to an optimal solution of a combinatorial problem, albeit in infinite computing time. Based on analogy with statistical mechanics, simulated annealing could be interpreted as a form of controlled random walk in the space of feasible solutions. The emergence of simulated annealing indicated that one could look for other ways to tackle combinatorial optimization problems and spurred the interest of the research community. In the following years, many other new approaches, mostly based on analogies with natural phenomena, were proposed such as tabu search, ant systems (Dorigo, 1992) and threshold methods (Dueck, 1990). Together with some older ones, in particular genetic algorithms (Holland, 1975), they gained an increasing popularity. Now collectively known under the name of meta-heuristics, a term originally coined by Glover (1986), these methods have become, over the last fifteen years, the leading edge of heuristic approaches for solving combinatorial optimization problems.

3.2 Tabu search

Fred Glover proposed in 1986 a new approach, which he called tabu search, to allow hill climbing to overcome local optima. In fact, many elements of this first tabu search proposal, and some elements of later elaborations, were already introduced in Glover (1977), including short term memory to prevent the reversal of recent moves, and longer term frequency memory to reinforce attractive components. The basic principle of tabu search is to pursue the search whenever a local optimum is encountered by allowing non-improving moves; cycling back to previously visited solutions is prevented by the use of memories, called tabu lists, that record the recent history of the search. The key idea to exploit information to guide the search can be linked to the informed search methods proposed in the late 70's in the field of artificial intelligence (Nilsson, 1980). It is interesting to note that, in 1986 also, Hansen proposed an approach similar to tabu search, which he named steepest ascent/mildest descent (Hansen, 1986). It is also important to remark that Glover did not see tabu search as a proper heuristic, but rather as a meta-heuristic, i.e., a general strategy for guiding and controlling "inner" heuristics specifically tailored to the problems at hand.

3.3 Search space and neighborhood structure

As we just mentioned, tabu search extends hill climbing methods. In fact, the basic tabu search can be seen as simply the combination of hill climbing with short-term memories. It follows that the two first basic elements of any tabu search heuristic are the definition of its search space and its neighborhood structure.

The search space is simply the space of all possible solutions that can be considered (visited) during the search. For instance, in the job shop scheduling problem of Section 2, the search space could simply be the set of feasible solutions to the problem, where each point in the search space corresponds to a set of m machine schedules that satisfies all the specified constraints. While in that case the definition of the search space seems quite natural, it is not always so. Consider now the capacitated plant location problem of Section 2: the feasible space involves both integer location and continuous flow variables that are linked by strict conditions; moreover, as has been indicated before, for any feasible set of values for the location variables, one can fairly easily retrieve optimal values for the flow variables by solving the associated transportation problem. In this context, one could obviously use as a search space the full feasible space; this would involve manipulating both location and flow variables, which is not an easy

task. A more attractive search space is the set of feasible vectors of location variables, i.e., feasible vectors in $\{0, 1\}^{|J|}$ (where $|J|$ is the cardinality of set J), any solution in that space being "completed" to yield a feasible solution to the original problem by computing the associated optimal flow variables. It is interesting to note that these two possible definitions are not the only ones. Indeed, on the basis of Remark 2 above, one could also decide to search instead the set of extreme points of the set of feasible flow vectors, retrieving the associated location variables by simply noting that a plant must be open whenever some flow is allocated to it. In fact, this type of approach was used successfully by Crainic, Gendreau and Farvolden (2000) to solve the fixed charge multi-commodity network design problem, which is a more general problem that includes the capacitated plant location problem as a special case. It is also important to note that it is not always a good idea to restrict the search space to feasible solutions. In many cases, allowing the search to move to infeasible solutions is desirable, and sometimes necessary (see subsection 4.3 for further details).

Closely linked to the definition of the search space is that of the neighborhood structure. At each iteration of tabu search, the local transformations that can be applied to the current solution, denoted S, define a set of neighboring solutions in the search space, denoted $N(S)$ (the neighborhood of S). Formally, $N(S)$ is a subset of the search space defined by:

$$N(S) = \{\textit{solutions obtained by applying a single local transformation to } S\}.$$

In general, for any specific problem at hand, there are many more possible (and even, attractive) neighborhood structures than search space definitions. This follows from the fact that there may be several plausible neighborhood structures for a given definition of the search space. This is easily illustrated on our job shop scheduling problem. In order to simplify the discussion, we assume in the following that the search space is the feasible space.

Simple neighborhood structures for the job shop scheduling problem are obtained by considering the sequence of jobs associated with a machine schedule, where the position of a job in the sequence corresponds to its processing order on the machine. For example, one can move a job at another position in the sequence or interchange the position of two jobs. While these neighborhood structures involve only one or two jobs, the neighborhoods they define contain all the feasible schedules that can be obtained from the current one either by moving any single job at any other position or by interchanging any two jobs. Examining these neighborhoods can thus be fairly demanding. In practice, it is often possible to reduce the computational burden, by identifying a restricted subset of moves that are feasible and can lead to improvements. We refer the interested reader to Vaessens, Aarts and Lenstra (1996) and Anderson, Glass and Potts (1997) for a more detailed discussion of these issues.

When different definitions of the search space are considered for a given problem, neighborhood structures will inevitably differ to a considerable degree. This can be illustrated on our capacitated plant location problem. If the search space is defined with respect to the location variables, neighborhood structures will usually involve the so-called "Add/Drop" and "Swap" moves that respectively change the status of one site (i.e., either opening a closed facility or closing an open one) and move an open facility from one site to another (this move amounts to performing simultaneously an Add move and a Drop move). If, however, the search space is the set of extreme points associated with feasible flow vectors, these moves become meaningless. One should instead consider moves defined by the application of pivots to the linear

programming formulation of the transportation problem, where each pivot operation modifies the flow structure to move the current solution to an adjacent extreme point.

The preceding discussion should have clarified a major point: choosing a search space and a neighborhood structure is by far the most critical step in the design of any tabu search heuristic. It is at this step that one must make the best use of the understanding and knowledge he/she has of the problem at hand.

3.4 Tabus

Tabus are one of the distinctive elements of tabu search when compared to hill climbing. As we already mentioned, tabus are used to prevent cycling when moving away from local optima through non-improving moves. The key realization here is that when this situation occurs, something needs to be done to prevent the search from tracing back its steps to where it came from. This is achieved by making certain actions tabu. This might mean not allowing the search to return to a recently visited point in the search space or not allowing a recent move to be reversed. For example, in the job shop scheduling problem, if a job j has been moved to a new position in a machine schedule, one could declare tabu moving that job back to its previous position for some number of iterations (this number is called the tabu tenure of the move).

Tabus are stored in a short-term memory of the search (the tabu list) and usually only a fixed and fairly limited quantity of information is recorded. In any given context, there are several possibilities regarding the recorded information. One could record complete solutions, but this requires a lot of storage and makes it expensive to check whether a potential move is tabu or not; it is therefore seldom used. The most commonly used tabus involve recording the last few transformations performed on the current solution and prohibiting reverse transformations (as in the example above); others are based on key characteristics of the solutions themselves or of the moves.

To better understand how tabus work, let us go back to our reference problems. In the job shop scheduling problem, one could define tabus in several ways. To continue our example where a job j has just been moved from position p_1 to position p_2, one could declare tabu specifically moving back j to position p_1 from position p_2 and record this in the short-term memory as the triplet (j, p_2, p_1). Note that this type of tabu will not constrain the search much, but that cycling may occur if j is then moved to another position p_3 and then from p_3 to p_1. A stronger tabu would involve prohibiting moving back j to p_1 (without consideration for its current position) and be recorded as (j, p_1). An even stronger tabu would be to disallow moving j at all, and would simply be noted as (j).

In the capacitated plant location problem, tabus on Add/Drop moves should prohibit changing the status of the affected location variable and can be recorded by noting its index. Tabus for Swap moves are more complex. They could be declared with respect to the site where the facility was closed, to the site where the facility was opened, to both locations (i.e., changing the status of both location variables is tabu), or to the specific swapping operation.

Multiple tabu lists can be used simultaneously and are sometimes advisable. For example, in the capacitated plant location problem, if one uses a neighborhood structure that contains both Add/Drop and Swap moves, it might be a good idea to keep a separate tabu list for each type of move.

Standard tabu lists are usually implemented as circular lists of fixed length. It has been shown, however, that fixed-length tabus cannot always prevent cycling, and some authors have proposed varying the tabu list length during the search (Glover, 1989, 1990, Skorin-Kapov, 1990, Taillard, 1990, 1991). Another solution is to randomly generate the tabu tenure of each move within some specified interval. Using this approach requires a somewhat different scheme for recording tabus, which are usually stored as tags in an array. The entries in this array typically record the iteration number until which a move is tabu. More details are provided in (Gendreau, Hertz and Laporte, 1994).

3.5 Aspiration criteria

While central to the tabu search method, tabus are sometimes too powerful. They may prohibit attractive moves, even when there is no danger of cycling, or they may lead to an overall stagnation of the search process. It is thus necessary to use algorithmic devices that will allow one to revoke (cancel) tabus. These are called aspiration criteria. The simplest and most commonly used aspiration criterion, found in almost all tabu search implementations, allows a tabu move when it results in a solution with an objective value better than that of the current best-known solution (since the new solution has obviously not been previously visited). Much more complicated aspiration criteria have been proposed and successfully implemented (see, for example, de Werra and Hertz (1989) and Hertz and de Werra (1991)), but they are rarely used. The key rule is that if cycling cannot occur, tabus can be disregarded.

3.6 A template for simple tabu search

We are now in the position to give a general template for tabu search, integrating the elements we have seen so far. We suppose that we are trying to minimize a function $f(S)$ (sometimes known as an objective or evaluation function) over some domain and we apply the so-called "best improvement" version of tabu search, i.e., the version in which one chooses at each iteration the best available move (even if this results in an increase in the function $f(S)$). This is the most commonly used version of tabu search.

Notation

- S, the current solution,
- S^*, the best-known solution,
- f^*, value of S^*,
- $N(S)$, the neighborhood of S,
- $\tilde{N}(S)$, the "admissible" subset of $N(S)$ (i.e., non-tabu or allowed by aspiration),
- T, tabu list.

Initialization

Choose (construct) an initial solution S_0.

Set $S := S_0$, $f^* := f(S_0)$, $S^* := S_0$, $T := \emptyset$.

Search

While *termination criterion not satisfied* do

- Select S in $\underset{S' \varepsilon \tilde{N}(S)}{argmin}\ [f(S')]$;
- if $f(S) < f^*$, then set $f^* := f(S)$, $S^* := S$;
- record tabu for the current move in T (delete oldest entry if necessary);

Endwhile.

In this algorithm, *argmin* returns the subset of solutions in $\tilde{N}(S)$ that minimizes f.

3.7 Termination criteria

One may have noticed that we have not specified in our template a termination criterion. In theory, the search could go on forever, unless the optimal value of the problem at hand is known beforehand. In practice, obviously, the search has to be stopped at some point. The most commonly used stopping criteria in tabu search are:

- after a fixed number of iterations (or a fixed amount of CPU time);
- after some number of consecutive iterations without an improvement in the objective function value (the criterion used in most implementations);
- when the objective function reaches a pre-specified threshold value.

3.8 Probabilistic tabu search and candidate lists

Normally, one must evaluate the objective function for every element of the neighborhood $N(S)$ of the current solution. This can be extremely expensive from a computational standpoint. In probabilistic tabu search, only a random sample $N'(S)$ of $N(S)$ is considered, thus significantly reducing the computational overhead. Another attractive feature is that the added randomness can act as an anti-cycling mechanism. This allows one to use shorter tabu lists than would be necessary if a full exploration of the neighborhood was performed. One the negative side, it is possible to miss excellent solutions (see subsection 6.3 for more detail). It is also possible to probabilistically select when to apply tabu criteria.

Another way to control the number of moves examined is by means of candidate list strategies, which provide more strategic ways of generating a useful subset $N'(S)$ of $N(S)$. In fact, the probabilistic approach can be considered to be one instance of a candidate list strategy, and may also be used to modify such a strategy. Failure to adequately address the issues involved in creating effective candidate lists is one of the more conspicuous shortcomings that differentiates a naive tabu search implementation from one that is more solidly grounded. Relevant designs for candidate list strategies are discussed in Glover and Laguna (1997). We also discuss a useful type of candidate generation approach in subsection 4.4.

4. EXTENSIONS TO THE BASIC CONCEPTS

Simple tabu search as described above can sometimes successfully solve difficult problems, but in most cases, additional elements have to be included in the search strategy to make it fully effective. We now briefly review the most important of these.

4.1 Intensification

The idea behind the concept of search intensification is that, as an intelligent human being would probably do, one should explore more thoroughly the portions of the search space that seem "promising" in order to make sure that the best solutions in these areas are found. In general, intensification is based on some intermediate-term memory, such as a recency memory, in which one records the number of consecutive iterations that various "solution components" have been present in the current solution without interruption. For instance, in the capacitated plant location problem, one could record how long each site has had an open facility. A typical approach to intensification is to restart the search from the best currently known solution and to "freeze" (fix) in it the components that seem more attractive. To continue with our capacitated plant location problem, one could freeze a number of facilities in sites that have been often selected in previous iterations and perform a restricted search on the other sites. Another technique that is often used consists of changing the neighborhood structure to one allowing more powerful or more diverse moves. In the capacitated plant location problem, if Add/Drop moves were used, Swap moves could be added to the neighborhood structure. In probabilistic tabu search, one could increase the sample size or switch to searching without sampling. Intensification is used in many tabu search implementations, although it is not always necessary. This is because there are many situations where the normal search process is thorough enough.

4.2 Diversification

One of the main problems of all methods based on local search, and this includes tabu search in spite of the beneficial impact of tabus, is that they tend to be too "local" (as their name implies), i.e., they tend to spend most, if not all, of their time in a restricted portion of the search space. The negative consequence of this fact is that, although good solutions may be obtained, one may fail to explore the most interesting parts of the search space and thus end up with solutions that are still far from the optimal ones. Diversification is an algorithmic mechanism that tries to alleviate this problem by forcing the search into previously unexplored areas of the search space. It is usually based on some form of long-term memory of the search, such as a frequency memory, in which one records the total number of iterations (since the beginning of the search) that various "solution components" have been present in the current solution or have been involved in the selected moves. For instance, in the capacitated plant location problem, one could record the number of iterations during which each site has had an open facility. In the job shop scheduling problem, one could note how many times each job has been moved. In cases where it is possible to identify useful "regions" of the search space, the frequency memory can be refined to track the number of iterations spent in these different regions.

There are two major diversification techniques. The first, called restart diversification, involves introducing a few rarely used components in the current solution (or the best known solution) and restarting the search from this point. In the capacitated plant location problem, one

could thus open one or more facilities at locations that have seldom been used up to that point and resume searching from that plant configuration (one could also close facilities at locations that have been used the most frequently). In the job shop scheduling problem, a job that has not occupied a particular position in a machine schedule can be forced to that position. The second diversification method, called continuous diversification, integrates diversification considerations directly into the regular searching process. This is achieved by biasing the evaluation of possible moves by adding to the objective a small term related to component frequencies. An extensive discussion on these two techniques is provided by Soriano and Gendreau (1996). A third way of achieving diversification is strategic oscillation which will be discussed in the next subsection.

We would like to stress that ensuring proper search diversification is possibly the most critical issue in the design of tabu search heuristics. It should be addressed with extreme care fairly early in the design phase and revisited if the results obtained are not up to expectations.

4.3 Allowing infeasible solutions

Accounting for all problem constraints in the definition of the search space often restricts the searching process too much and can lead to mediocre solutions. In such cases, constraint relaxation is an attractive strategy, since it creates a larger search space that can be explored with "simpler" neighborhood structures. Constraint relaxation is easily implemented by dropping selected constraints from the search space definition and adding to the objective, weighted penalties for constraint violations. In the capacitated plant location problem, this can be done by allowing solutions with flows that exceed the capacity of one or more plants. This, however, raises the issue of finding correct weights for constraint violations. An interesting way of circumventing this problem is to use self-adjusting penalties, i.e., weights are adjusted dynamically on the basis of the recent history of the search. Weights are increased if only infeasible solutions were encountered in the last few iterations, and decreased if all recent solutions were feasible; see (Gendreau, Hertz and Laporte, 1994) for further details. Penalty weights can also be modified systematically to drive the search to cross the feasibility boundary of the search space and thus induce diversification. This technique, known as strategic oscillation, was introduced in (Glover, 1977) and used since in several successful tabu search procedures. An important early variant oscillates among alternative types of moves, hence neighborhood structures, while another oscillates around a selected value for a critical function.

4.4 Surrogate and auxiliary objectives

There are many problems for which the true objective function is quite costly to evaluate, a typical example being the capacitated plant location problem when one searches the space of location variables. Remember that, in this case, computing the objective value for any potential solution entails solving the associated transportation problem. When this occurs, the evaluation of moves may become prohibitive, even if sampling is used. An effective approach to handle this issue is to evaluate neighbors using a surrogate objective, i.e., a function that is correlated to the true objective, but is less computationally demanding, in order to identify a small set of promising candidates (potential solutions achieving the best values for the surrogate). The true objective is then computed for this small set of candidate moves and the best one selected to become the new current solution. An example of this approach is found in (Crainic et al., 1993).

Another frequently encountered difficulty is that the objective function may not provide enough information to effectively drive the search to more interesting areas of the search space. A typical illustration of this situation is observed when the fixed costs for open plants in the capacitated plant location problem are much larger than the transportation costs. In this case, it is indicated to open as few plants as possible. It is thus important to define an auxiliary objective function to orient the search. Such a function must measure in some way the desirable attributes of the solutions. In our example, one could use a function that would favor, for the same number of open plants, solutions with plants having just a small amount of flow, thus increasing the likelihood of closing them in subsequent iterations. It should be noted that developing an effective auxiliary objective is not always easy and may require a lengthy trial and error process. In some other cases, fortunately, the auxiliary objective is obvious for anyone familiar with the problem at hand (see the work of Gendreau, Soriano and Salvail (1993) for an illustration).

5. SOME PROMISING AREAS FOR FUTURE APPLICATION

The concepts and techniques described in the previous sections are sufficient to design effective tabu search heuristics for many combinatorial problems. Most early tabu search implementations, several of which were extremely successful, relied indeed almost exclusively on these algorithmic components (Friden, Hertz and de Werra, 1989, Hertz and de Werra, 1987, Skorin-Kapov, 1990, Taillard, 1991). Nowadays, however, most leading edge research in tabu search makes use of more advanced concepts and techniques. While it is clearly beyond the scope of an introductory tutorial, such as this one, to review this type of advanced material, we would like to give readers some insight into it by briefly describing some current trends. Readers who wish to learn more about this topic should read our survey paper (Gendreau, 2002) and some of the references provided in this section.

A large part of the recent research in tabu search deals with various techniques for making the search more effective. These include methods for better exploitation of the information that becomes available during search and creating better starting points, as well as more powerful neighborhood operators and parallel search strategies. For more details, see the taxonomy of Crainic, Toulouse and Gendreau (1997), and the survey of Cung et al. (2002). The numerous techniques for utilizing the information are of particular significance since they can lead to dramatic performance improvements. Many of these rely on elite solutions (the best solutions previously encountered) or on parts of these to create new solutions, the rationale being that "fragments" of excellent solutions are often identified quite early in the search process. However, the challenge is to complete these fragments or to recombine them (Glover, 1992, Glover and Laguna, 1993, Glover and Laguna, 1997, Rochat and Taillard, 1995). Other methods, such as the reactive tabu search of Battiti and Tecchiolli (1994), are aimed at finding ways to move the search away from local optima that have already been visited.

Another important trend (this is, in fact, a pervasive trend in the whole meta-heuristics field) is hybridization, i.e., using tabu search in conjunction with other solution approaches such as genetic algorithms (Crainic and Gendreau, 1999, Fleurent and Ferland, 1996), lagrangean relaxation (Grünert, 2002), constraint programming (Pesant and Gendreau, 1999), and column generation (Crainic, Gendreau and Farvolden, 2000). A whole chapter on this topic is found in (Glover and Laguna, 1997). Problem specific information and simple heuristics can also be used

in conjunction with different components of tabu search. For example, in Burke et al. (1998), problem specific heuristics are used to realize diversification.

The literature on tabu search has also started moving away from its traditional application areas (graph theory problems, scheduling, vehicle routing) to new ones: continuous optimization (Rolland, 1996), multi-objective optimization (Gandibleux et al. 2000), stochastic programming (Lokketangen and Woodruff, 1996), mixed integer programming (Crainic, Gendreau and Farvolden, 2000, Lokketangen and Glover, 1996), real-time decision problems (Gendreau et al., 1999), etc. These new areas confront researchers with new challenges that, in turn, call for novel and original extensions of the method.

6. TRICKS OF THE TRADE

6.1 Getting started

Newcomers to tabu search, trying to apply the method to a problem that they wish to solve, are often confused about what they need to do to come up with a successful implementation. Basically, they do not know where to start. We believe that the following step-by-step procedure will help and provides a useful framework for getting started.

A step-by-step procedure

1. **Read** one or two good introductory papers to gain some knowledge of the concepts and of the vocabulary (see the references provided in "Sources of Additional Information" at the end).
2. **Read** several papers describing in detail applications in various areas to see how the concepts have been actually implemented by other researchers (see the references provided in "Sources of Additional Information" at the end).
3. **Think** a lot about the problem at hand, focusing on the definition of the **search space** and the **neighborhood structure**.
4. **Implement** a **simple** version based on this search space definition and this neighborhood structure.
5. **Collect statistics** on the performance of this simple heuristic. It is usually useful at this point to introduce a variety of **memories**, such as frequency and recency memories, to really track down what the heuristic does.
6. **Analyze results** and **adjust** the procedure accordingly. It is at this point that one should eventually introduce mechanisms for search intensification and diversification or other intermediate features. Special attention should be paid to **diversification**, since this is often where simple tabu search procedures fail.

6.2 More tips

In spite of carefully following the procedure outlined above, it is possible to end up with a heuristic that produces mediocre results. If this occurs, the following tips may prove useful:

1. If there are **constraints**, consider **penalizing the violation of** them. Letting the search move to infeasible solutions is often necessary in highly constrained problems to allow for a meaningful exploration of the search space (see Section 4).
2. Reconsider the **neighborhood structure** and change it if necessary. Many tabu search implementations fail because the neighborhood structure is too simple. In particular, one should make sure that the chosen neighborhood structure allows for a sensible evaluation of possible moves (i.e., the moves that seem intuitively to move the search in the "right" direction should be the ones that are likely to be selected); it might also be a good idea to introduce a **surrogate objective** (see Section 4) to achieve this.
3. **Collect more statistics**. For example, recording the number and quality of previously visited local optima can be useful to find a good trade-off between intensification and diversification
4. **Follow the execution of the algorithm step-by-step** on some reasonably sized instances (e.g., is the algorithm behaving as expected on particular solution configurations? is the algorithm converging prematurely? etc.).
5. Reconsider **diversification**. As mentioned earlier, this is a critical feature in most tabu search implementations.
6. **Experiment** with **parameter** settings. Many tabu search procedures are extremely sensitive to parameter settings; it is not unusual to see the performance of a procedure dramatically improve after changing the value of one or two key parameters (unfortunately, it is not always obvious to determine which parameters are the key ones in a given procedure).

6.3 Additional tips for probabilistic tabu search

While probabilistic tabu search is an effective way of tackling many problems, it creates difficulties of its own that need to be carefully addressed. The most important of these occurs because more often than not, the best solutions returned by probabilistic tabu search will not be local optima with respect to the neighborhood structure being used. This is particularly annoying since, when it happens, better solutions can be easily obtained, sometimes even manually. An easy way to address this is to simply perform a local improvement phase (using the same neighborhood operator) from the best found solution at the end of the tabu search itself. One could alternately switch to tabu search without sampling (again from the best found solution) for a short duration before completing the algorithm. A possibly more effective technique is to add, throughout the search, an intensification step without sampling. This will mean that the best solutions available in the various regions of the space explored by the method will be found and recorded. This is similar to the method proposed by Glover and Laguna (1993). They employed special aspiration criteria for allowing the search to reach local optima at useful junctures.

6.4 Parameter calibration and computational testing

Parameter calibration and computational experiments are key steps in the development of any algorithm. This is particularly true in the case of tabu search, since the number of parameters required by most implementations is fairly large and the performance of a given procedure can vary quite significantly when parameter values are modified. The first step in any serious

computational experimentation is to select a good set of benchmark instances (either by obtaining them from other researchers or by constructing them), preferably with some reasonable measure of their difficulty and with a wide range of size and difficulty. This set should be split into two subsets, the first one being used at the algorithmic design and parameter calibration steps, and the second reserved for performing the final computational tests that will be reported in the paper(s) describing the heuristic under development. The reason for doing so is quite simple: when calibrating parameters, one always run the risk of overfitting, i.e., finding parameter values that are excellent for the instances at hand, but poor in general, because these values provide too good a "fit" (from the algorithmic standpoint) to these instances. Methods with several parameters should thus be calibrated on much larger sets of instances than ones with few parameters to ensure a reasonable degree of robustness. The calibration process itself should proceed in several stages:

1. Perform exploratory testing to find good ranges of parameters. This can be done by running the heuristic with a variety of parameter settings.
2. Fix the value of the parameters that appear to be "robust", i.e., which do not seem to have a significant impact on the performance of the algorithm.
3. Perform systematic testing for the other parameters. It is usually more efficient to test values for only a single parameter at a time, the others being fixed at what appear to be reasonable values. One must be careful, however, for cross effects between parameters. For example, assume that value x_1 for parameter p_1 leads to good results when the other parameters are fixed at their default values, and that value x_2 for parameter p_2 leads to good results when the other parameters are fixed at their default values. Then, it might happen that value x_1 for parameter p_1 **and** value x_2 for parameter p_2 lead to poor results. Where such effects exist, it can be important to jointly test pairs or triplets of parameters, which can be an extremely time-consuming task.

The paper by Crainic et al. (1993) provides a detailed description of the calibration process for a fairly complex tabu search procedure and can used as a guideline for this purpose.

7. CONCLUSION

Tabu search is a powerful algorithmic approach that has been applied with great success to many difficult combinatorial problems. A particularly nice feature of tabu search is that it can quite easily handle the "dirty" complicating constraints that are typically found in real-life applications. It is thus a really practical approach. It is not, however, a panacea: every reviewer or editor of a scientific journal has seen more than his/her share of failed tabu search heuristics. These failures stem from two major causes: an insufficient understanding of fundamental concepts of the method (and we hope that this tutorial will help in alleviating this shortcoming), but also, more often than not, a crippling lack of understanding of the problem at hand. One cannot develop a good tabu search heuristic for a problem that he/she does not know well! This is because significant problem knowledge is absolutely vital to perform the most basic steps of the development of any tabu search procedure, namely the choice of a search space and the choice of an effective neighborhood structure. If the search space and/or the neighborhood structure are inadequate, no amount of tabu search expertise will be sufficient to save the day. A last word of caution: to be successful, all meta-heuristics need to achieve both depth and breadth in their searching process; depth is usually not a problem for tabu search, which is quite aggressive in this

respect (it generally finds pretty good solutions very early in the search), but breadth can be a critical issue. To handle this, it is extremely important to develop an effective diversification scheme.

SOURCES OF ADDITIONAL INFORMATION

- Good introductory papers on tabu search may be found in (Glover and Laguna, 1993, Glover, Taillard and de Werra, 1993, Hertz and de Werra, 1991, de Werra and Hertz, 1989) and, in French, in (Soriano and Gendreau, 1997).
- The book by Glover and Laguna (1997) is the ultimate reference on tabu search. Apart from the fundamental concepts of the method, it presents a considerable amount of advanced material, as well as a variety of applications. It is interesting to note that this book contains several ideas applicable to tabu search that yet remain to be fully exploited.
- Two issues of *Annals of Operations Research* devoted respectively to "Tabu Search" (Glover, Laguna, Taillard and de Werra, 1993) and "Metaheuristics in Combinatorial Optimization" (Laporte and Osman, 1996) provide a good sample of applications of tabu search.
- The books made up from selected papers presented at the Meta-Heuristics International Conferences (MIC) are also extremely valuable. At this time, the books for the 1995 Breckenridge conference (Osman and Kelly, 1996), the 1997 Sophia-Antipolis one (Voss, Martello, Osman and Roucairol, 1999) and the 1999 Angra dos Reis one (Ribeiro and Hansen, 2002) have been published. The proceedings of MIC'2001, held in Porto, are currently available electronically on the website located at the URL: http://tew.ruca.ua.ac.be/eume/MIC2001.
- Two books of interest have recently been published. The one edited by Glover and Kochenberger (2003) addresses meta-heuristics in general. The other, edited by Pardalos and Resende (2002), has a broader scope but contains a nice chapter on metaheuristics.

ACKNOWLEDGEMENTS

The authors are grateful to the Canadian Natural Sciences and Engineering Council and the Quebec Fonds FCAR for their financial support.

REFERENCES

Anderson E.J., Glass C.A. and Potts C.N. (1997), "Machine Scheduling", in *Local Search in Combinatorial Optimization*, E.H.L. Aarts and J.K. Lenstra (eds.), John Wiley & Sons, pp. 361-414.

Battiti R. and Tecchiolli G. (1994), "The Reactive Tabu Search", *ORSA Journal on Computing 6*, 126-140.

Burke E., De Causmaecker P. and Vanden Berghe G. (1998), "A Hybrid Tabu Search Algorithm for the Nurse Rostering Problem", Selected Papers from the 2nd Asia Pacific Conference on Simulated Evolution and Learning, *Lecture Notes in Artificial Intelligence 1585*, pp. 187-194.

Crainic T.G. and Gendreau M. (1999), "Towards an Evolutionary Method – Cooperative Multi-Thread Parallel Tabu Search Heuristic Hybrid", in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I.H. Osman and C. Roucairol (eds.), Kluwer Academic Publishers, pp. 331-344.

Crainic T.G., Gendreau M. and Farvolden J.M. (2000) "Simplex-based Tabu Search for the Multicommodity Capacitated Fixed Charge Network Design Problem", *INFORMS Journal on Computing 12*, 223-236.

Crainic T.G., Gendreau M., Soriano P. and Toulouse M. (1993), "A Tabu Search Procedure for Multicommodity Location/Allocation with Balancing Requirements", *Annals of Operations Research 41*, 359-383.

Crainic T.G., Toulouse M. and Gendreau M. (1997), "Toward a Taxonomy of Parallel Tabu Search Heuristics", *INFORMS Journal on Computing 9*, 61-72.

Cung V.-D., Martins S.L., Ribeiro C.C. and Roucairol C. (2002), "Strategies for the Parallel Implementation of Metaheuristics", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds.), Kluwer Academic Publishers, pp. 263-308.

Dorigo M. (1992), "Optimization, Learning and Natural Algorithms", Ph.D. Dissertation, Departimento di Elettronica, Politecnico di Milano, Italy.

Dueck G. and Scheuer T. (1990), "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing", *Journal of Computational Physics 90*, 161-175.

Fleurent C. and Ferland J.A. (1996), "Genetic and Hybrid Algorithms for Graph Colouring", *Annals of Operations Research 63*, 437-461.

Friden C., Hertz A. and de Werra D. (1989), "STABULUS: A Technique for Finding Stable Sets in Large Graphs with Tabu Search", *Computing 42*, 35-44.

Gandibleux X., Jaszkiewicz A., Freville A., Slowinski R. (eds.) (2000), *Journal of Heuristics*, volume 6, number 3, Special Issue: Multiple Objective Metaheuristics, 291-431.

Gendreau M. (2002), "Recent Advances in Tabu Search", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds.), Kluwer Academic Publishers, pp. 369-377.

Gendreau M., Guertin F., Potvin J.-Y. and Taillard É.D. (1999), "Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching", *Transportation Science 33*, 381-390.

Gendreau M., Hertz A. and Laporte G. (1994), "A Tabu Search Heuristic for the Vehicle Routing Problem", *Management Science 40*, 1276-1290.

Gendreau M., Soriano P. and Salvail L. (1993), "Solving the Maximum Clique Problem Using a Tabu Search Approach", *Annals of Operations Research 41*, 385-403.

Glover F. (1977), "Heuristics for Integer Programming Using Surrogate Constraints", *Decision Sciences* 8, 156-166.

Glover F. (1986), "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers and Operations Research 13*, 533-549.

Glover F. (1989), "Tabu Search – Part I", *ORSA Journal on Computing 1*, 190-206.

Glover F. (1990), "Tabu Search – Part II", *ORSA Journal on Computing 2*, 4-32.

Glover F. (1992), "Ejection chains, Reference Structures and Alternating Path Methods for Traveling Salesman Problems", University of Colorado. Shortened version published in *Discrete Applied Mathematics 65*, 223-253, 1996.

Glover F. and Kochenberger G.A. (eds.) (2003), Handbook of Metaheuristics, Kluwer Academic Publishers, Boston.

Glover F. and Laguna M. (1993), "Tabu Search", in C.R Reeves (ed.), in *Modern Heuristic Techniques for Combinatorial Problems*, C.R. Reeves (ed.), Blackwell, pp. 70-150.

Glover F. and Laguna M. (1997), *Tabu Search*, Kluwer Academic Publishers, Boston.

Glover F., Laguna M., Taillard É.D. and de Werra D. (eds.) (1993), "Tabu Search", *Annals of Operations Research 41*, J.C. Baltzer Science Publishers, Basel, Switzerland.

Glover F., Taillard É.D. and de Werra D. (1993), " A User's Guide to Tabu Search", *Annals of Operations Research 41*, 3-28.

Grünert T. (2002), "Lagrangean Tabu Search", in *Essays and Surveys in Metaheuristics*, C.C. Ribeiro and P. Hansen (eds.), Kluwer Academic Publishers, pp. 379-397.

Hansen P. (1986), "The steepest ascent miidest descent heuristic for combinatorial programming", *Congress on Numerical Methods in Combinatorial Optimization*, Capri, Italy.

Hertz A. and de Werra D. (1987), "Using Tabu Search for Graph Coloring", Computing *39*, 345-351.

Hertz A. and de Werra D. (1991), "The Tabu Search Metaheuristic: How We Used It", *Annals of Mathematics and Artificial Intelligence 1*, 111-121.

Holland J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

Kirkpatrick S., Gelatt Jr. C.D. and Vecchi M.P. (1983), "Optimization by Simulated Annealing", *Science 220*, 671-680.

Laporte G. and Osman I.H. (eds.) (1996), "Metaheuristics in Combinatorial Optimization", *Annals of Operations Research 63*, J.C. Baltzer Science Publishers, Basel, Switzerland.

Lokketangen A. and Glover F. (1996), "Probabilistic Move Selection in Tabu Search for 0/1 Mixed Integer Programming Problems", in *Meta-Heuristics: Theory and Applications*, I.H. Osman and J.P. Kelly (eds.), Kluwer Academic Publishers, pp. 467-488.

Lokketangen A. and Woodruff D.L. (1996), "Progressive Hedging and Tabu Search applied to Mixed Integer (0,1) Multistage Stochastic Programming", *Journal of Heuristics 2*, 111-128.

Nilsson N.J. (1980), *Principles of Artificial Intelligence*, Morgan Kaufmann, Los Altos, CA.

Osman I.H. and Kelly J.P. (eds.) (1996), *Meta-Heuristics: Theory and Applications*, Kluwer Academic Publishers, Boston.

Pardalos P.M. and Resende M.G.C. (eds.) (2002), *Handbook of Applied Optimization*, Oxford University Press, New-York.

Pesant G. and Gendreau M. (1999), "A Constraint Programming Framework for Local Search Methods", *Journal of Heuristics 5*, 255-280.

Ribeiro C.C. and Hansen P. (eds.) (2002), *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers, Norwell, MA.

Rochat Y. and Taillard É.D. (1995), "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing", *Journal of Heuristics 1*, 147-167.

Rolland E. (1996), "A Tabu Search Method for Constrained Real-Number Search: Applications to Portfolio Selection", Working Paper, The Gary Anderson Graduate School of Management, University of California, Riverside.

Skorin-Kapov J. (1990), "Tabu Search Applied to the Quadratic Assignment Problem", *ORSA Journal on Computing 2*, 33-45.

Soriano P. and Gendreau M. (1996), "Diversification Strategies in Tabu Search Algorithms for the Maximum Clique Problems", *Annals of Operations Research 63*, 189-207.

Soriano P. and Gendreau M. (1997), "Fondements et applications des méthodes de recherche avec tabous", *RAIRO (Recherche opérationnelle) 31*, 133-159.

Taillard É.D. (1990), "Some efficient heuristic methods for the flow shop sequencing problem", *European Journal of Operational Research 47*, 65-74.

Taillard É.D. (1991), "Robust taboo search for the quadratic assignment problem", *Parallel Computing 17*, 443-455.

Vaessens R.J.M., Aarts E.H.L. and Lenstra J.K. (1996), "Job Shop Scheduling by Local Search", *INFORMS Journal on Computing 8*, 302-317.

Voss S., Martello S., Osman I.H. and Roucairol C. (eds.) (1999), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer Academic Publishers, Boston.

de Werra D. and Hertz A. (1989), "Tabu Search Techniques: A Tutorial and an Application to Neural Networks", *OR Spektrum 11*, 131-141.